

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

_____ О.І. Ролік

«__» _____ 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра
з напрямку підготовки 6.050103 «Програмна інженерія»
на тему: «Телеграм-бот побудови маршруту з використанням громадського
транспорту міста Києва»**

Виконав :

студент IV курсу, групи ІТ-51

Шородок Володимир Володимирович _____

Керівник:

Доцент Писаренко Андрій Володимирович _____

Рецензент: _____

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2019 рік

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки – 6.050103 «Програмна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.І. Ролік

«___» _____ 2019 р.

ЗАВДАННЯ
на дипломний проект студенту
Шородку Володимиру Володимировичу

1. Тема проекту «Telegram-бот побудови маршруту з використанням громадського транспорту міста Києва» керівник проекту доцент кафедри АУТС Писаренко Андрій Володимирович, затверджені наказом по університету від «___» _____ 2019 р.
№ _____

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту

Операційна система Windows, мови програмування Python та Javascript, середовище програмування Microsoft Visual Studio, цільова платформа Telegram, обрана для розробки технологія – Telebot, CherryPy, Requests, СУБД – MongoDB.

4. Зміст пояснювальної записки

1. Вступ 2. Проектування 3. Аналіз вимог 4. Аналіз предметної області 5 Огляд існуючих рішень 6.Розробка telegram-бота. 7. База даних – MongoDB 8.Telegram 9. Тестування системи 10. Опис власної реалізації. 11. висновки

Додатки:

5. Код програми

6. Перелік графічного матеріалу

UML Діаграма варіантів використання, UML Діаграма активності телеграм-боту, UML Діаграма класів оюроки повідомлення, UML Діаграма послідовності обробки користувацького запиту.

7. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Вибір тематичного напрямку та узгодження теми дипломного проекту	22.02.2019	
2	Аналіз теоретичних матеріалів та вивчення предметної області	15.04.2019	
3	Розробка технічного завдання, вибір методів та засобів реалізації задачі	24.04.2019	
4	Огляд існуючих рішень з тематики роботи	27.04.2019	
5	Розробка структури прототипу та проектування системи	06.05.2019	
6	Реалізація проекту	20.05.2019	
7	Налагодження та перевірка програми	23.05.2019	
8	Оформлення пояснювальної записки	03.06.2019	
9	Передзахист дипломного проекту	04.06.2019	
10	Доопрацювання пояснювальної записки та підготовка презентації	18.06.2019	
11	Захист дипломного проекту	20.06.2019	

Студент

Шородок В.В.

Керівник проекту

Писаренко А.В.

АНОТАЦІЯ

Шородок В.В. Телеграм-бот побудови маршруту з використанням громадського транспорту міста Києва КПІ ім. Ігоря Сікорського, Київ, 2019.

Пояснювальна записка дипломного проекту містить 15 таблиць, 1 додаток, 12 рисунків та 21 джерело – загалом 66 сторінок.

Об'єкт дослідження: чат-бот побудови маршруту з використанням громадського транспорту міста Києва.

Мета дипломного проекту: підвищення зручності користування громадським транспортом за допомогою розроблення телеграм-боту, який будує маршрути з використанням громадського транспорту м. Києва.

КЛЮЧОВІ СЛОВА: БОТ, ЧАТ-БОТ, РОБОТ, ТЕЛЕГРАМ-БОТ, PYTHON, МЕССЕНДЖЕР, TELEGRAM, МАРШРУТИ, ГРОМАДСЬКИЙ ТРАНСПОРТ.

SUMMARY

Shroodok V.V. Telegram of the construction of a route using public transport of the city of Kyiv Igor Sikorsky KPI, Kyiv, 2019.

Explanatory note of the diploma project contains 15 tables, 1 application, 12 figures and 21 sources - a total of 66 pages.

Object of research: chat-bot of construction of a route using public transport of the city of Kiev.

The purpose of the diploma project: to increase the convenience of using public transport through the development of telegrams, which build routes using public transport in Kiev.

KEYWORDS: BOT, CHAT-BOT, ROBOT, TELEGRAM-BOT, PYTHON, MESSENGER, TELEGRAM, ROADS, PUBLIC TRANSPORT.

ЗМІСТ

ЗМІСТ	2
ВСТУП	4
1 ПРОЕКТУВАННЯ.....	7
2 АНАЛІЗ ВИМОГ	8
3 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	11
4.1. Додаток Eway.....	11
4.2. Synbot	12
5 РОЗРОБКА TELEGRAM-БОТА	14
5.1 Реєстрація чат-бота для Telegram Bot API	14
5.2 Розміщення чат-бота на віддаленому сервері Heroku	15
5.3 Середовище розробки.....	16
5.4 Мова програмування – Python	18
5.4.1 Модуль TeleBot.....	19
5.4.2 Модуль CherryPy	20
5.4.3 Модуль Requests	21
5.5 REST API, HTTP-запити, JSON та черга повідомлень.....	21
5.5.1 REST API.....	21
5.5.2 HTTP-запити.....	22
5.5.3 Формат JSON	24
5.5.4 Платформа Heroku	27
5.5.5 Черга повідомлень	28
5.6 Telegram Bot API.....	29
6 БАЗА ДАНИХ – MONGODB.....	46
7 TELEGRAM	50

					IT51.320БАК.001 ТП			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шородок В.В.			Телеграм-бот для побудови маршруту з використанням громадського транспорту. Відомість технічного проекту	Літ.	Арк.	Акрушів
Перевір.		Писаренко А.В.					2	61
Реценз.						КПІ ім. Ігоря Сікорського ФІОТ, гр. ІТ-51		
Н. Контр.		Шинкевич М.К.						
Затверд.								

8 ТЕСТУВАННЯ СИСТЕМИ.....	52
9 ОПИС ВЛАСНОЇ РЕАЛІЗАЦІЇ.....	57
ВИСНОВКИ.....	59

ВСТУП

Оскільки сучасний світ не стоїть на місці, технології стрімко розвиваються та кожного дня з'являються нові тренди та ідеї, багато з них мають можливість вплинути на ІТ галузь, яка в свою чергу впливає на інші галузі, що в подальшому впливатиме на наше повсякденне життя. Якщо до недавнього часу популярними були додатки або комп'ютерні програми, то на даний момент лідерство займають чат-боти, які мають великі перспективи в різних сферах нашого життя.

Для початку потрібно визначити, що представляє собою чат-бот. Чат-бот – це спеціалізований додаток, що дозволяє користувачам взаємодіяти зі сторонніми сервісами, якщо існує така необхідність і все це виконано через всім відомий інтерфейс чату.

Чат-бот – це деякий помічник, який спілкується з користувачами через повідомлення і має множину певних функцій. Тобто, можна отримати певну інформацію, написавши чат-боту спеціальну команду, яку в свою чергу останній інтерпретує певним чином. Так можна швидко переводити, коментувати, знаходити, тестувати, шукати, навчати, транслювати, вбудовуватися в інші сервіси і платформи, взаємодіяти з датчиками і речами, підключеними до інтернету.

AI (Artificial intelligence) в загальному використовується для скорочення витрат і підвищення якості послуг. Найбільш популярними напрямками розвитку AI є розпізнавання голосу і текстове спілкування за допомогою різних пошукових ботів. Наприклад можна згадати про всім відомі спливаючі віконця "консультантів" на багатьох сайтах. Однак, про якість таких консультацій можна посперечатися, але тенденція очевидна і від неї не збираються відступати. Яскравим прикладом автоматизації можна назвати величезну кількість ботів для прийому заявок на доставку їжі, бронювання столиків в ресторанах, розсилки реклами і багато іншого.

					ІТ51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		4

Такі боти дозволяють збільшити прибуток компаній, оскільки бот може обробляти заявки з більшою швидкістю, ніж людина, і зменшити навантаження на робочий персонал.

Кожний день мільйони людей користуються громадським транспортом міста Києва, досить часто виникає необхідність швидко дізнатися про найближчі зупинки, який транспорт зупиняється на них, коли прибуде на зупинку та побудувати оптимальні маршрути.

Також в місті Києві, галузь громадського транспорту розвивається, на більшості маршруток, трамваїв та тролейбусів уже встановлені GPS – маячки, які дають можливість відстежувати місцезнаходження транспорту та розраховувати час прибуття на ту, чи іншу зупинку.

Тому, для розв’язання певних потреб наведених раніше, щоб зберегти час людей та зробити планування маршруту легкою справою, було вирішено створити телеграм-бот побудови маршрутів з використанням громадського транспорту в місті Києві на платформі Telegram.

Telegram – це додаток, розроблений на мові програмування C ++, що дозволяє обмінюватися повідомленнями і файлами більшості форматів. Месенджер використовує спеціально розроблену серверну частину з закритим кодом, які працюють на серверах Німеччини і США. Telegram має ряд переваг:

- а) конфіденційність – всі чати зашифровуються, а повідомлення знищуються через зазначений час;
- б) швидкість – швидкість доставки повідомлень вище, ніж у аналогів;
- в) оптимізація – сервера Telegram розташовані по всьому світу, що підвищує надійність;
- г) відкритість – використання відкритого протоколу;
- д) відсутність підписок і реклами;
- е) відсутність обмежень на розмір повідомлень і вкладених файлів.

Актуальність бакалаврської роботи зумовлена високою популярністю месенджерів і таких засобів автоматизації як чат-боти серед користувачів мережі Інтернет. Чат-боти дозволяють спростити щоденні рутинні завдання, такі як отримання інформації про погоду, пробки, останні новини та інші. Головним достоїнством щодо класичних додатків є можливість заміщення всіх можливостей на платформі одного месенджера.

Було поставлено мету: створити телеграм-бота, який зможе відповідно до місцезнаходження користувачів, надати йому необхідну інформацію про найближчі зупинки, який саме громадський транспорт зупиняється на отриманих зупинках, за необхідністю повідомляти коли прибуде на зупинку необхідний транспорт, будувати оптимальні маршрути відповідно до запиту користувача.

Виходячи з поставленої мети, були поставлені наступні завдання:

- а) аналіз обраної предметної області;
- б) порівняння наявних аналогів чат-ботів;
- в) вибір технологій і середовища розробки;
- г) розробка чат-бота на платформі Telegram.

1 ПРОЕКТУВАННЯ

Проектування невід'ємна частина розробки будь-якої програмної системи. Умовно, цей процес можна розділити на кілька послідовних, пов'язаних один з одним етапів: формулювання цілей (збір вимог), аналіз предметної області та створення прототипу системи. Аналіз предметної області має на увазі більш глибоке вивчення вимог, що пред'являються до функціонала проектованої системи, опрацювання сценаріїв взаємодії з системою і алгоритмів в бізнес-логіці.

Чітко сформульовані цілі і завдання, ретельно проведений аналіз вимог, опрацювання сценаріїв взаємодії з системою і проектування бізнес-логіки - все це ключ успіху в досягненні необхідного результату. Таким чином, проектування збільшує шанси успішного складання проекту, а також економить час і гроші, адже внесення змін на початковому етапі розробки не так критично, як, наприклад, на етапі налагодження вже розробленої системи.

					IT51.320БАК.002 ПЗ	Лист
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

2 АНАЛІЗ ВИМОГ

Завданням дипломного проекту було визначено створення телеграм-боту для побудови маршрутів з використанням громадського транспорту міста Києва.

Перед початком проектування і розробки телеграм бота потрібно визначити основні вимоги та описати їх. Крім цього, потрібно проаналізувати предметну область та розглянути існуючі рішення, щоб переконатися в доцільності розробки. Оскільки систему, що розробляється планується впроваджувати в уже існуючий програмний комплекс, це накладає певні обмеження на архітектуру і технології, які можна буде використовувати.

Виходячи з поставлених цілей, потрібна розробка телеграм бота, який:

а) буде знати про всі можливі зупинки громадського транспорту та тип громадського транспорту, який зупиняється на них;

б) буде зберігати отримані дані в спеціальній базі даних, яка буде оптимізована для зберігання метрик і тимчасових даних;

в) буде певним чином зберігати маршрути та графік роботи громадського транспорту;

г) дозволить відстежувати поточне місцезнаходження користувача та громадського транспорту;

д) дозволить отримувати інформацію про прибуття громадського транспорту на зупинку;

е) дозволить будувати оптимальні маршрути руху з використанням громадського транспорту;

ж) дозволить знаходити найближчі до місцезнаходження користувача зупинки громадського транспорту.

А тепер детально опишемо кожен із пунктів. Користувачу потрібно буде знати про всі можливі зупинки громадського транспорту, відповідно до вказаного адресу.

					IT51.320БАК.002 ПЗ	Лист
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

Оскільки зупинки громадського транспорту змінюються не часто, ми можемо зберігати інформацію про них в базі даних, щоб не витратити час і ресурси для нового пошуку. Аналогічно ми можемо зробити з інформацією про маршрути та графіки роботи громадського транспорту.

Також користувач захоче бачити графік прибуття громадського транспорту на зупинку, його тип та маршрут. Для того, щоб знаходити зупинки громадського транспорту, які знаходяться неподалік користувача, ми повинні отримати доступ від користувача на відслідковування його місцезнаходження.

Користувачу надається можливість ввести потрібну йому адресу або вибрати необхідну зупинку і в залежності від вибраної адреси або зупинки, телеграм-бот буде для користувача оптимальний маршрут пересування по місту Києву з використанням громадського транспорту.

					IT51.320БАК.002 ПЗ	Лист
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

3 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Проаналізувавши всі можливі варіанти створення даного чат-бота було виявлено, що для написання такого бота необхідно використовувати одну з мов серверного програмування: Ruby, Node.JS, Python, PHP. Необхідно визначити яка саме мова найкраще підходить для написання такого бота. Також є важливим вміння працювати з REST (Representational State Transfer) API (Application Programming Interface), які надають месенджери, а в даному випадку це – TelegramBotAPI. Після цього важливо визначитися з типом бота:

а) боти, що мають здібність до навчання (тобто розуміють природні мови), які вміють використовувати логіку при спілкуванні з користувачем та обробляти природну мову для створення відповіді на повідомлення;

б) заскриптовані боти (боти, які не здатні зрозуміти природню мову), у них діалог з користувачем це заздалегідь сформований набір слів або речень, а «скрипт» це своєрідне дерево рішень, в якому сценарій, який був запрограмований заздалегідь є відповіддю на питання користувача. Діалоги в них в більшості випадків лінійні і структуровані.

І на останок, потрібно визначити мету бота, оскільки у нього повинна бути певна мета, тому що в іншому випадку він не матиме сенсу.

4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

4.1. Додаток Eway

Під час детального вивчення предметної області, було знайдено додаток “Eway” та відповідний веб-сайт.

Оскільки додатки починають втрачати популярність, велика частина людей починає використовувати телеграм-боти в повсякденному житті, я вирішив обрати тему: телеграм-бот для побудови маршруту з використанням громадського транспорту в місті Києві. Тому для того щоб створити телеграм-бота, який матиме можливості web-сайту та додатку потрібно визначити основні можливості додатку та web-сайту.

Додаток " Eway " підійде насамперед для того, щоб відстежувати потрібний вам транспорт на маршруті. Ви зможете вийти з дому і дійти до зупинки якраз до часу прибуття. Єдиний мінус - поки ще не всі транспортні засоби забезпечені GPS-маячками, тому їх рух не можна побачити на карті. Але що стосується Києва, майже всі маршрути «Київпастрансу» та великих транспортних компаній вже синхронізовані з додатком, правда, частина з них не відображається в режимі реального часу.

Користуватися додатком нескладно: потрібно включити GPS на своєму пристрої, і далі Eway сам встановить ваше місто за замовчуванням і знайде найближчі зупинки. Серед них вибирається потрібна, далі - відповідний маршрут зі списку транспорту. Додаток показує очікуваний час прибуття транспорту конкретного маршруту на зупинку на закладці з загальною інформацією, а також інформацію про інтервал руху транспорту, часу його роботи і вартості проїзду. У підменю «Карта» видно, де саме знаходиться потрібний вам тролейбус / трамвай / маршрутка.

Серед інших корисних функцій варто відзначити прокладання маршрутів з промальовуванням на карті міста (метрополітен теж враховується). До речі, зараз сервіс працює в меншій кількості міст України, ніж спочатку, через конфлікти з компаніями, що встановили права на

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		11

передачу GPS-даних. Відстеження маршрутів є в Києві, Дніпрі, Кам'янському, Полтаві, Дрогобичі, Хмельницькому, Кропивницького, Львові.

Серед переваг даного додатку можна виділити такі: користувач може отримати необхідну інформацію про громадський транспорт, його місцезнаходження, час прибуття, найближчі зупинки та можливість побудувати маршрут.

Основними недоліками є:

- а) відсутність GPS-маячків на усіх транспортних засобах;
- б) повільна швидкість роботи;
- в) відсутність телеграм-бота.

Оскільки не існує телеграм-бота, який був би пов'язаний з нашою предметною областю, ми розглянемо телеграм бот «Синоптик»@synbot.

4.2. Synbot

Synbot – це бот, головна мета якого полягає у інформуванні користувача щодо останнього прогнозу погоди. Цей бот має можливість надавати інформацію про прогноз погоди не тільки на поточний день, а й на декілька наступних.

Кожного ранку люди цікавляться прогнозом погоди. Інколи для отримання останнього прогнозу погоди за допомогою браузера або додатку не вистачає часу. Synbot мінімізує витрати часу на отримання прогнозу погоди, достатньо відкрити додаток Telegram та увійти в чат з ботом, який без сумніву допоможе визначитись з гардеробом. Саме завдяки швидкодії цей бот стає все більш популярнішим.

Розглянемо основні функції даного бота. Як тільки ми відкриваємо діалог з ботом, користувачу одразу надається можливість дізнатися прогноз погоди на поточний день. Далі бот дає змогу вибрати параметри відображення прогнозу погоди, такі як: період часу, місто, максимальна та

мінімальна температура повітря, переважний напрямок вітру та можливість опадів.

Якщо є необхідність користувач може зберегти раніше введені параметри та налаштувати автоматичні оповіщення, тобто повідомлення від бота з прогнозом погоди.

Переваги Synbot:

- а) зручність інтерфейсу;
- б) можливість регулювати та зберігати параметри;
- в) швидкодія;
- г) простий алгоритм дії;
- д) автосповіщення..

					IT51.320БАК.002 ПЗ	Лист
						13
Ізм.	Лист	№ докум.	Підпис	Дата		

5 РОЗРОБКА TELEGRAM-БОТА

5.1 Реєстрація чат-бота для Telegram Bot API

Першим кроком для розробки бота є реєстрація у спеціального чат-бота «BotFather». Реєстрація починається з команди «/ newbot», після чого пропонується ввести назву чат-бота з обов'язковою умовою: в кінці назва має бути зазначено «Bot» або «_bot». Якщо всі умови були виконані, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) і URL-адресу для доступу з чат-боту.

Для установки додаткових параметрів, таких як іконка чат-бота, вітальне повідомлення, опис чат-бота, а так само видалення наявних чат-боти, існують такі команди (таблиця 5.1.1).

Таблиця 5.1.1 – Доступні команди для зміни чат-ботів

Команда	Опис
/ setname	Змінює існуюче ім'я
/ setdescription	Привласнює текст, який буде відображатися при першому відкритті чата бота
/ setabouttext	Присвоює текст в поле «Про чат-боті»
/ setuserpic	Присвоює обрану картинку
/ setcommands	дозволяє створити список доступних команд
/ deletebot	Видаляє обраного чат-бота

Крім команд для зміни основних параметрів чат-бота існує ряд команд, які дозволяють виводити незмінні параметри (токен), а так же привласнювати значення, які представлені в таблиці 5.1.2.

Таблиця 5.1.2. – Доступні команди для додаткової настройки чат-бота

Команда	Опис
/ token	Повертає отриманий раніше токен у обраного бота
/ revoke	Анулює токен доступу до боту
/ setinline	Вмикає або вимикає можливість викликати бота з інших чатів
/ setinlinegeo	Вмикає або вимикає можливість передавати розташування бота з іншого чату
/ setinlinefeedback	Дозволяє отримувати інформацію про кількість обраних користувачами команд
/ setjoininggroup	Визначає чи може бути доданий в групові діалоги бот
/ setprivacy	Включає режим конфіденційності. В цьому режимі бот отримує, обробляє і відсилає назад інформацію окремо для кожного користувача в чаті

Після налаштувань на стороні Telegram і отриманні токена можна приступити до розробки програмної частини чат-бота.

5.2 Розміщення чат-бота на віддаленому сервері Heroku

Для постійного та швидкого отримання доступу до нашого чат-боту, після розробки його необхідно розмістити на локальному сервері. В якості платформи був обраний Heroku. Розміщення на хмарної PaaS-платформі Heroku схоже з роботою розподіленої системи контролю версій (Git).

Існує три способи розгортання:

- а) Heroku Git і Container Registry використовуючи Heroku CLI;
- б) підключення акаунта GitHub з автоматичним розгортанням;
- в) підключення акаунта Dropbox з автоматичним розгортанням.

У роботі був обраний спосіб і використанням Heroku CLI.

а) для розгортання чат-боти були виконані наступні дії:

б) реєстрація на хмарну SaaS-платформі Heroku;

в) завантажений і встановлений Heroku CLI;

г) в терміналі операційної системи або вбудованому в IDE виконаний вхід в акаунт Heroku командою «heroku login»;

д) клонований репозиторій з віддаленого Git-сервера Heroku на локальну машину за допомогою команди «heroku git: clone -a <APP_NAME>»;

е) зафіксовані зміни в коді за допомогою команд «git add.» і «git commit -am <COMMIT_NAME>»;

ж) всі зафіксовані зміни були відправлені на віддалений сервер хмарних обчислень Heroku командою «git push heroku master».

Після виконання всіх зазначених команд в правильній послідовності починається передача даних на віддалені сервера. Якщо розгортання пройшло успішно, то на терміналі відобразиться довідкова інформація про стан програми та режим доступу до нього.

Таким чином, чат-бот був розгорнутий на віддаленому сервері хмарних обчислень Heroku, що дозволить користувачам отримувати доступ безперебійно.

5.3 Середовище розробки

Visual Studio Code Visual Studio Code - це зрозумілий, лаконічний, але багатофункціональний редактор вихідного коду, який сумісний з Windows, MacOS і Linux.

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		16

IDE підтримує значну кількість популярних мов програмування. Підтримувані розширення і середовища виконання представлені нижче:

- а) JavaScript;
- б) TypeScript;
- в) Node.js;
- г) C ++;
- д) C #;
- е) Java;
- ж) Python;
- з) PHP;
- и) Go;
- к) .NET;
- л) Unity.

Visual Studio Code включає в себе вбудований відладчик, для зручності розробки програмних продуктів, інструменти для роботи з Git-репозиторіями, підсвічування синтаксису, засоби для рефакторинга і IntelliSense (технологія автодоповнення, яка запропонує команду за першими літерами). Visual Studio Code був заснований на фреймворку Electron, що дозволяє використовуючи Node.js розробляти настільні додатки, що працюють на двигуні Blink.

5.4 Мова програмування – Python

Python – це мова програмування загального призначення, яка націлена перш за все на збільшення продуктивності самого розробника ПЗ, ніж коду, який він пише.

Якщо говорити простою мовою, Python дає можливість написати практично все: веб- / настільні додатки, ігри, автоматизовані інформаційні системи, комплексні системи, системи управління життєзабезпеченням і багато іншого. Крім того, поріг входження низький, а код багато в чому небагатослівний і зрозумілий навіть для того, хто ніколи на ньому не писав. За рахунок простоти коду, майбутній супровід програм, які були написані на Python, стає легше і приємніше по відношенню до Java або C. А з точки зору бізнесу це дає скорочення витрат і підвищення ефективності трудових ресурсів.

Практично на всіх платформах і операційних системах був реалізований інтерпретатор Python, що є безперечною перевагою. Першою такою мовою був C, однак його типи даних на різних машинах досить часто займали різну кількість пам'яті, а це представляє деяку перешкоду при написанні великих програм.

Також, важлива риса – розширюваність мови, цьому надається велике значення і, як пише сам автор Гвідо ван Россум, мова була задуманий саме як розширювана. Це означає, що є можливість вдосконалення мови усіма зацікавленими розробниками. Інтерпретатор був розроблений на мові програмування C і вихідний код доступний для будь-яких змін.

У разі необхідності, можна вставити його в свою програму і використовувати як власну вбудовану оболонку. Або ж, написавши на C доповнення до Python і скомпілювавши програму, отримати "розширений" інтерпретатор з новими можливостями.

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		18

Наступна значна перевага – наявність великого числа модулів, які можна підключити, що забезпечують різні додаткові можливості. Такі модулі зазвичай пишуться на С, на самому Python і можуть бути створені більш досвідченими програмістами. Як приклад можна навести такі модулі:

а) Numerical Python – надає програмісту більш розширені математичні можливості, такі як маніпуляції з цілими векторами і матрицями;

б) Tkinter – дає змогу будувати додатки з використанням графічного інтерфейсу користувача (GUI);

в) OpenGL – це велика бібліотека графічного моделювання дво- та тривимірних об'єктів Open Graphics Library.

Єдиним недоліком, на який звертав увагу сам автор є порівняно невисока швидкість виконання Python програми. Однак, це не відіграє велику роль у порівнянні з перевагами мови при написанні програм не дуже критичних до швидкості виконання. Із використаних модулів можна виділити: TeleBot, CherryPy, Requests, які більш детально будуть описані нижче.

5.4.1 Модуль TeleBot

Модуль TeleBot – є оболонкою над запитами до TelegramBotAPI, використовується для спрощення і мінімізації написаного коду. Всі типи визначені в types.py. Всі вони повністю відповідають визначенням типів API Telegram, за винятком from поля Message, яке перейменовано в from_user (оскільки from це зарезервований токен Python). До таких атрибутів, як message_id можна звертатися безпосередньо, наприклад: message.message_id. При написанні програм не дуже вимогливих до швидкості виконання варто звернути увагу, що атрибут message.chat може належати як певному користувачеві, так і до групового чату, це з повністю окупається перевагами мови.

В класі TeleBot розташовані всі методи API. Щоб слідувати загальним угодам про імена Python, вони перейменовані. Наприклад: sendMessage, editMessageText, edit_message_text. Функція, прикрашена декоратором примірника TeleBot – обробник повідомлень. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає True або False для певного повідомлення і обробник отримує дозвіл на обробку повідомлення, якщо повертається True.

5.4.2 Модуль CherryPy

Написаний на мові програмування Python, об'єктно-орієнтований веб-фреймворк це – CherryPy. Створений для швидкої розробки веб-додатків для мережі Інтернет. Являє собою надбудову над HTTP-протоколом.

Модуль може виступати в якості самостійного веб-сервера або працювати під управлінням іншої серверної програми на основі протоколу WSGI. CherryPy не призначений для: обробки шаблонів для виведення даних, отримання доступу до бази даних, авторизації користувача. Фреймворк стає розширеним за рахунок фільтрів, що представляють собою прості інтерфейси, які викликаються в певних точках процесу обробки запитів / відповідей і складаються з семи функцій. Головним компонентом TurboGears є модуль CherryPy.

Процес встановлення CherryPy зовсім нескладний, достатньо просто завантажити його і виконати кілька коротких кроків з файлу README. Модуль CherryPy встановлюється так, як і будь-який інший модуль Python, тому не потрібно вказувати каталог установки. Якщо порівнювати з іншими, більш складними серверними технологіями, модуль CherryPy завжди є доступним для використання, як і будь-який інший модуль Python, якщо він буде імпортований за допомогою інструкції import. В реальності CherryPy являє собою не більше ніж самостійний додаток на мові Python, який вміє керувати своїм власним багатопотоковим веб-сервером, завдяки цьому ми

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		20

маємо змогу виконати «сценарій на стороні сервера» досить просто, як запустити просту команду у вікні терміналу. За замовчуванням сервер запускається локально на 8080 порту, і одразу виконує функцію, яку йому передали, але так само можна задати певні налаштування вручну, такі як: IP-адреса, порт, відкритий і закритий ключі SSL.

5.4.3 Модуль Requests

Requests – бібліотека Python, яка виконує HTTP-запити (HyperText Transfer Protocol). Починаючи від передачі параметрів в URL-адресах до відправки користувацьких заголовків і перевірки SSL.

5.5 REST API, HTTP-запити, JSON та черга повідомлень

5.5.1 REST API

REST – це стиль архітектури програмного забезпечення, який використовується для побудови розподілених масштабованих веб-сервісів, які використовують HTTP запити.

REST (Representational State Transfer) – стиль взаємодії компонентів розподіленого додатку в мережі. REST має узгоджений набір обмежень, що враховуються при проектуванні розподіленої гіпермедіа-системи. У визначених випадках (інтернет-магазини, пошукові системи) це спричинює підвищення продуктивності та змінення архітектури, а саме її спрощення. Компоненти в REST своєю взаємодією нагадують взаємодію клієнта і сервера в Інтернет.

Виклик віддаленої процедури в мережі Інтернет може представляти собою звичайний HTTP-запит (зазвичай «GET» або «POST»; такий запит називають «REST-запит»), а всі необхідні для передачі дані передаються в якості параметрів запиту.

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		21

Для веб-служб, які були побудовані з урахуванням REST, застосовують так званий термін «RESTful».

На відміну від веб-сервісів на основі SOAP, не існує єдиного офіційного стандарту для терміну RESTful веб-API. Оскільки REST є архітектурним стилем, в той час як SOAP є протоколом. REST не є стандартом сам по собі і не зважаючи на це, більшість RESTful-реалізацій використовують відомі стандарти, такі як: HTTP, URL, JSON і XML.

5.5.2 HTTP-запити

Протокол Передачі Гіпертексту (HTTP) – це один із відомих протоколів стеку TCP / IP (Transmission Control Protocol / Internet Protocol), з самого початку він був розроблений для розміщення і отримання HTML (HyperText Markup Language) сторінок і на даний момент використовується для розподілених інформаційних систем. HTTP широко використовується у Інтернет, а саме для передачі інформації і являє собою найбільш використовуваний прикладний протокол.

HTTP це протокол, який визначається типом запит / відповідь. Коли клієнт або веб браузер, відправляє повідомлення з запитом на сервер, HTTP протокол самостійно визначає типи надісланих серверу повідомлень, які використовуються клієнтом для запиту до веб сторінки, а також типи повідомлень, які сервер використовує для відповіді. Три найбільш популярні типи повідомлень це: GET, POST і PUT. На рисунку 5.5.2.1 наведено приклад GET-запиту.

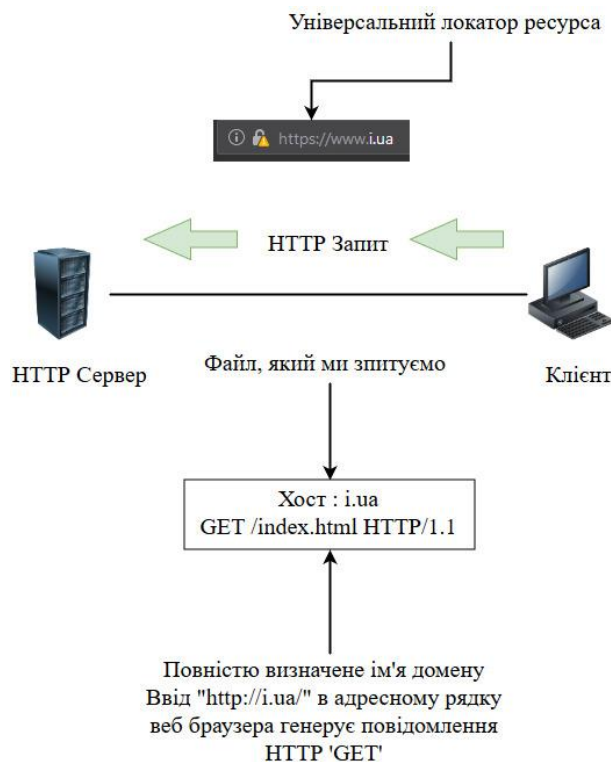


Рисунок 5.5.2.1 – HTTP протокол з використанням GET

Для того, щоб відправити повідомлення, які завантажують дані на веб сервер використовуються POST і PUT. Наприклад, коли користувач вводить дані в форму, яка знаходиться на веб сторінці, POST додає ці дані в повідомлення і вони посилаються на сервер. PUT завантажує ці ресурси або інший контент на веб сервер.

HTTP протокол є надзвичайно гнучким, але це не робить його безпечним протоколом. Повідомлення POST відправляють дані на сервер у вигляді звичайного тексту, який може бути з легкістю перехоплений. Аналогічно, відповіді, які надходять від сервера, зазвичай також не зашифровані.

Для того, щоб створити безпечну комунікацію через Інтернет використовується безпечний HTTP протокол (HTTPS), він дозволяє отримувати доступ або публікувати інформацію на веб сервері. HTTPS дає змогу використовувати аутентифікацію і шифрування, щоб захистити дані від перехоплення, коли вони переміщуються між клієнтом і сервером. HTTPS

визначає деякі свої правила для проходження даних між прикладним і транспортним рівнями.

5.5.3 Формат JSON

JSON (JavaScript Object Notation) – це формат для обміну даними, комфортний для зберігання, читання та написання як людиною, так і комп'ютером. Він був заснований на підмножині відомої мови програмування JavaScript та визначений в стандарті ECMA-262 3rd Edition - December 1999. JSON – представляє собою майже звичайний текстовий формат, повністю незалежний від мови реалізації, але він використовує деякі правила, які знайомі програмістам С-подібних мов, таких як С, С ++, С # , Java, JavaScript, Perl, Python і багатьох інших. Саме ці властивості роблять JSON ідеальною мовою, яку можна використовувати для обміну даними.

JSON з самого початку був заснований на двох структурах даних:

а) колекція пар ключ / значення. Ця концепція, у різних мовах може бути реалізована як об'єкт, запис, структура, словник, хеш, іменованний список або асоціативний масив;

б) упорядкований список значень. Більшість мов програмування реалізують це як масив, вектор, список або послідовність. Це універсальні структури даних. Майже всі сучасні мови програмування мають змогу підтримувати їх в будь-якій формі. Досить легко припустити, що формат даних, який зберігаються в JSON, незалежний від мови програмування та повинен бути заснований на структурах вказаних вище.

В нотації JSON це виглядає так:

а) об'єкт – це неупорядкований набір пар ключ/значення. Об'єкт починається з відкритої фігурної дужки і закінчується закритою фігурною дужкою. Кожне ім'я в об'єкті супроводжується двокрапкою, а пари ключ/значення розділяються комою (рисунок 5.5.3.1);

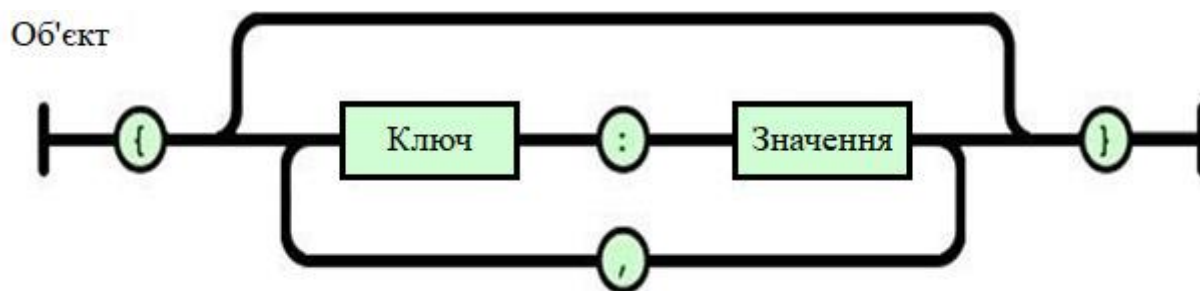


Рисунок 5.5.3.1 – Схематичне представлення об'єкта JSON

б) масив – це впорядкована колекція, яка може містити різні значення. Масив починається з квадратної дужки і закінчується цією дужкою. Значення в масиві розділяються комою (рисунок 5.5.3.2);



Рисунок 5.5.3.2 - Схематичне представлення масиву JSON

в) значення – може бути числом, рядком в подвійних лапках, true, false, null, об'єктом або масивом. Ці структури можуть бути вкладеними (рисунок 5.5.3.3);

г) рядок – це так званий набір символів в кодуванні Unicode, укладений в подвійні лапки, або символ, який використовує зворотний слеш в якості символу екранування. Символ представляється як один символний рядок. Схожий синтаксис використовується в мовах програмування C і Java (рисунок 5.5.3.4);

Число в JSON передається так, як в C або Java, крім того, що використовується тільки десяткова система числення (рисунок 5.5.3.5).

Значення

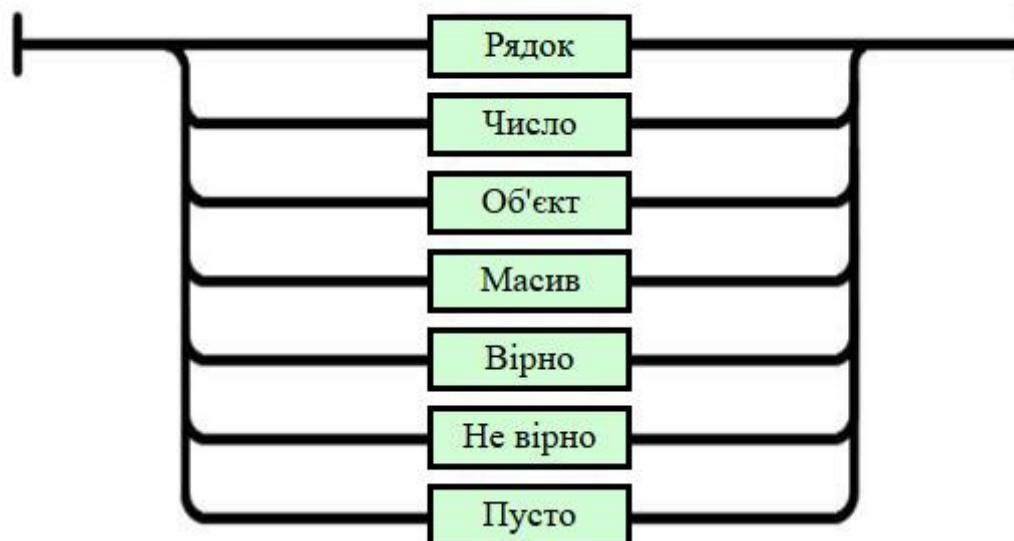


Рисунок 5.5.3.3 – Схематичне представлення значення в JSON

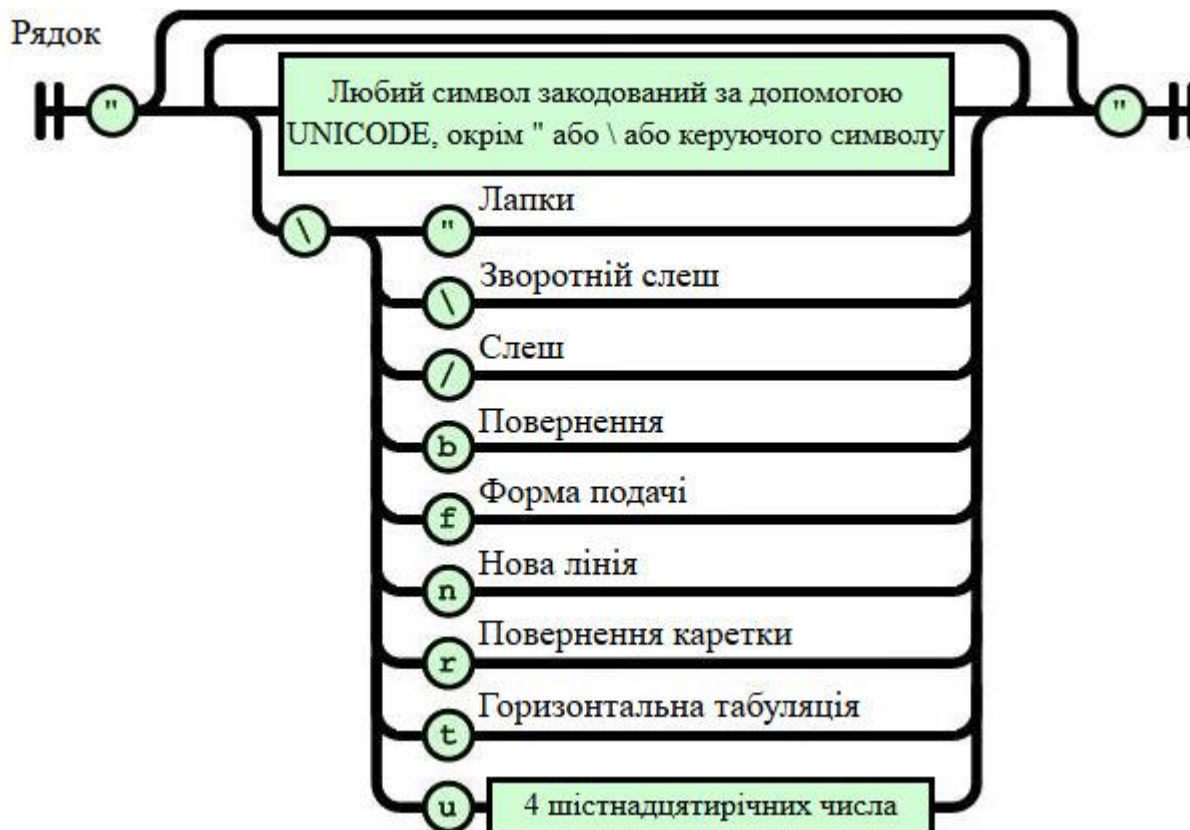


Рисунок 5.5.3.4 - Схематичне представлення рядки в JS

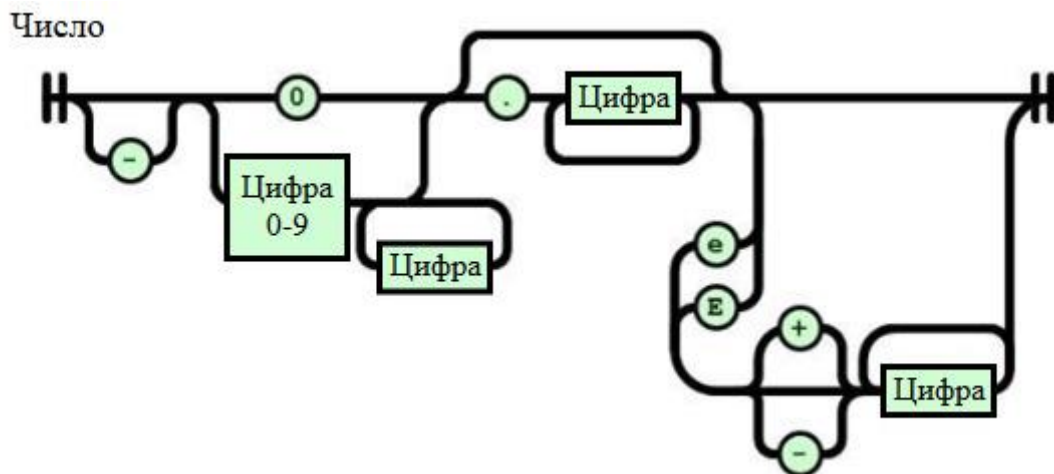


Рисунок 5.5.3.5 – Схематичне представлення числа в JSON

5.5.4 Платформа Heroku

Heroku – представляє собою хмарну PaaS-платформу, яка підтримує велику кількість мов програмування. З 2010 року вона стає дочірньою компанією Salesforce. Heroku, це одна з перших хмарних платформ, яка з'явилася в червні 2007 року і в початковій версії підтримувала тільки одну мову програмування Ruby, але на даний момент список підтримуваних мов розширився і включає в себе Java, Node.js, Scala, Clojure, Python, Go і PHP. Зазвичай на серверах використовуються операційні системи Debian або Ubuntu. У 2011 році була впроваджена підтримка Node.js і Clojure. Також платформа має змогу підтримувати такі СУБД, як Cloudant, Membase, MongoDB і Redis, окрім PostgreSQL. Додатки, які працюють на даній хмарній платформі, використовують також DNS-сервер (зазвичай програми мають особисте доменне ім'я, яке виглядає так – «назва проекту.herokuapp.com»).

Для всіх програм виділяються декілька незалежних віртуальних процесів, які мають назву – «dynos». Вони розміщені по спеціальній віртуальній сітці, яка складається з декількох серверів. Однією з особливостей Heroku є те, що вона має власну систему контролю версій.

5.5.5 Черга повідомлень

Черга повідомлень (message queue, MQ) – це архітектура обміну повідомленнями між різними компонентами в програмних системах в асинхронному режимі. Це дозволяє, відправляти повідомлення одним компонентом-відправником системи в один час, а отримати та обробити іншим компонентом-одержувачем в зовсім інший. Системи такого типу складаються з producer'a (відправника) та consumer'a (одержувача), які взаємодіють між собою за допомогою постачальника (broker), який по-іншому можна визначити, як деякий MQ-сервер.

Черга – це структура даних з обмеженим доступом до елементів, яку можна описати, як «перший прийшов-перший вийшов». Обмеженість полягає в тому, що елемент можна додати тільки в кінець черги, а вибрати елемент тільки з початку, при вилученні елемента з черги, він знищується.

Особливості використання MQ-архітектури:

а) слабке зв'язування – створюються неявні інтерфейси які обмінюються даними, дозволяють процесам бути незалежними один від одного та дозволяють зберігати бажаний формат повідомлень;

б) надмірність – дає можливість уникати нераціонального використання ресурсів системи або мережі, в свою чергу зберігаючи ще необроблену інформацію;

в) масштабованість – за допомогою розподілу процесів обробки інформації дає можливість збільшувати продуктивність MQ-сервера;

г) еластичність і можливість витримувати пікові навантаження – при високому навантаженні на додаток, черги повідомлень можуть служити, так званим буфером для накопичення інформації, дозволяючи змінити швидкість обробки інформації і, тим самим, знизити загальне навантаження на систему або мережу;

д) стійкість – дає можливість відокремити один процес від одного і отримання повідомлень, завдяки цьому, в разі відмови процесу-обробника, надається можливість відновити робочий стан системи, відкладанням обробки повідомлення;

е) гарантована доставка – повідомлення в будь-якому випадку будуть оброблені та доставлені, незалежно від робочого стану системи-відправника та системи-одержувача повідомлень. Це досягається використанням асинхронного зв'язку і можливістю зберігання повідомлення на MQ-сервері, поки воно не буде оброблено;

ж) гарантований і одноразовий порядок доставки – значна частина MQ-серверів дає змогу доставити дані саме в тому порядку, в якому вони були відправлені, при цьому гарантує, що після того, як повідомлення буде прочитано, воно буде видалено з черги;

з) буферизація – це гарантія доставки повідомлень з максимальною ефективністю, яка досягається за допомогою незалежності швидкості обробки повідомлень від швидкості їх відправки, це в свою чергу відбувається завдяки використанню вище згаданої структури черги – своєрідного буфера між системою-обробником і системою-відправником.

5.6 Telegram Bot API

Bot API представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Кожен бот – це спеціально створений акаунт для автоматичного оброблення та відправлення повідомлень.

Документація Telegram Bot API виділяє два максимально протилежних способи отримання оновлень:

- а) періодичні запити;
- б) установка веб-хуків.

Вхідні оновлення зберігаються до того моменту, поки сервер не обробить його, але не більш ніж 24 години. У відповідь отримуємо об'єкт

					IT51.320БАК.002 ПЗ	Лист
						29
Ізм.	Лист	№ докум.	Підпис	Дата		

Update, який серіалізовано в JSON, незалежно від способу отримання оновлень.

Перший і найбільш простий варіант полягає в періодичному опитуванні серверів Telegram на наявність нової інформації. Відкривається з'єднання на нетривалий час і всі оновлення відразу відправляються боту, все це здійснюється через зв'язок Long Polling. Цей спосіб є простим, але не дуже надійним.

Веб-хуки працюють трохи інакше. Якщо в чат приходить повідомлення, то Telegram сам говорить про це, в цьому і полягає робота вебхука. Тепер відпадає необхідність періодично опитувати сервери, тим самим, зникає причина помилок пошукових роботів. Однак за цю можливість, потрібно платити необхідністю установки повноцінного веб-сервера на той пристрій, на якому і планується запускати пошукових ботів.

Так само для коректної роботи необхідно мати власний сертифікат-SSL (Secure Sockets Layer), оскільки веб-хуки в Telegram працюють тільки по HTTPS (рисунок 5.6.1).

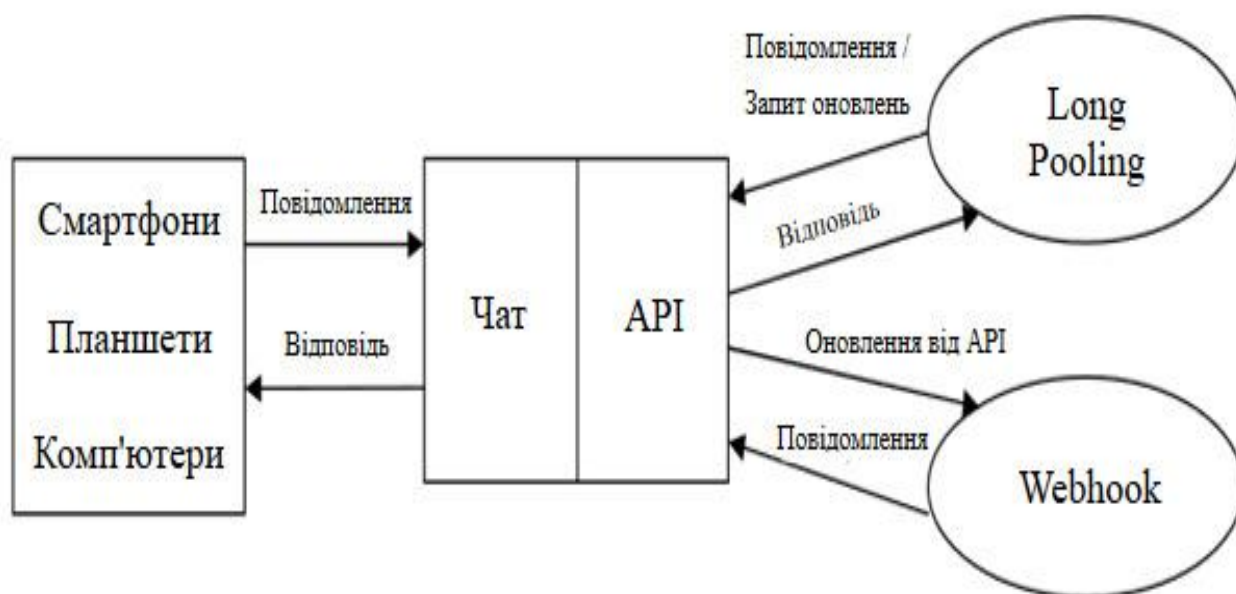


Рисунок 5.6.1 – Принцип роботи чат-бота на платформі Telegram

Для отримання токена необхідно написати спеціальному боту @BotFather.

Нижче опишемо приклади доступних методів для API:

а) `getUpdates` – це метод, який використовується для отримання оновлень за технологією `long polling`;

б) `setWebhook` – цей метод прив'язує до боту `url` домену, де міститься поточний запущений бот;

в) `sendMessage` – метод, який відправляє текстове повідомлення в клієнтській частині Telegram;

г) `sendLocation` – метод, який відправляє повідомлення з координатами місцезнаходження пристрою;

д) `getFile` – метод, який повертає вкладені файли по їх імені. Допускаються `POST` і `GET` запити.

Існують 4 способи передачі параметрів в Bot API:

а) запит в `URL`;

б) `application / x-www-form-urlencoded`;

в) `application / json`;

г) `multipart / form-data` (підходить для завантаження файлів).

Для роботи з Telegram Bot API була вивчена документація, в якій описані всі методи і параметри, було встановлено, що всі відповіді приходять в `JSON`-форматі. В ході написання чат-бота були протестовані і використані наступні методи і типи:

Метод `getUpdates` використовується для отримання оновлень через `long polling`. Відповідь повертається у вигляді масиву об'єктів `Update`. Параметри методу наведені в таблиці 5.6.1.

Таблиця 5.6.1 – Параметри getUpdates

Параметри	Тип	обов'язковий	Опис
Offset	Integer	Ні	Зсув відносно початку масиву Update, дозволяє переглядати нові повідомлення, ігноруючи старі.
Limit	Integer	Ні	Обмежує кількість елементів масиву Update. Приймає значення від 1 до 100.
Timeout	Integer	Ні	Таймаут в секундах для тривалого опитування. За умовчанням 0, тобто так званий короткий запит.

Метод setWebhook необхідний для завдання URL веб-хука, на який бот буде відправляти оновлення. Кожного разу при отриманні оновлення на цю адресу буде відправлений HTTPS POST з серіалізованим в JSON об'єктом Update.

При помилці запиту до сервера спроба буде повторена декілька разів. Для більшої безпеки рекомендується включити токен в URL веб-хука, наприклад, так: [https://yourwebhookserver.com/ <token>](https://yourwebhookserver.com/<token>). Ми можемо бути впевнені, що запити до веб-хуку відправляє саме Telegram, оскільки ніхто сторонній не знає токена. Параметри даного методу наведені в таблиці 5.6.2

Таблиця 5.6.2 – Параметри setWebhook

Параметри	Тип	обов'язковий	Опис
url	String	Ні	HTTPS url для відправки запитів.
Sertificate	InputFile	Ні	Завантаження публічного ключа для перевірки кореневого сертифіката.

Метод `getWebhookInfo` містить в собі інформацію про поточний стан веб-хука. Параметри цього методу наведені в таблиці 5.6.3.

Таблиця 5.6.3 – Параметри які повертає `getWebhookIn`

Параметри	Тип	Опис
<code>url</code>	String	url веб-хука, може бути порожнім.
<code>has_custom_certificate</code>	Boolean	True, тільки якщо веб-хук використовує свій самозавірений сертифікат.
<code>pending_update_count</code>	Integer	Кількість оновлень, які очікують своєї черги.
<code>last_error_date</code>	Integer	Не обов'язково. Unix-час останньої помилки доставки оновлення на вказаний веб-хук.
<code>last_error_message</code>	String	Не обов'язково. Опис останньої помилки доставки оновлення на вказаний веб-хук.

Метод `sendMessage` використовується для відправлення повідомлень. Параметри методу наведені в таблиці 5.6.5.

Таблиця 5.6.4 – Параметри `sendMessage`

Параметри	Тип	Обов'язковий	Опис
<code>chat_id</code>	Integer або String	Так	Унікальний ідентифікатор цільового чату або імені цільового каналу.
<code>Text</code>	String	Так	Текст повідомлення, яке необхідно відправити.

Продовження таблиці 5.6.4.

Параметри	Тип	Обов'язковий	Опис
parse_mode	String	Ні	Потрібно надіслати Markdown або HTML, щоб додатки Telegram відображали напівжирний, курсивний, текст з фіксованою шириною або вбудовані URL-адреси в повідомленні бота.
disable_web_page_preview	Boolean	Ні	Відключає попередній перегляд посилань в повідомленні.
disable_notification	Boolean	Ні	Відправляє повідомлення тихо. Користувачі iOS не отримують повідомлення, користувачі Android отримують повідомлення без звуку.
reply_to_message_id	Integer	Ні	Ідентифікатор вихідного повідомлення.
reply_markup	InlineKeyboardMarkup або ReplyKeyboardMarkup або ReplyKeyboardHide	Ні	Розширені можливості пошуку інтерфейсу. Являє собою JSON-серіалізований об'єкт для вбудованої клавіатури, яка призначена для користувача.

Метод sendPhoto використовується для відправки фото. Параметри методу наведені в таблиці 5.6.5.

Таблиця 5.6.5 – Параметри sendPhoto

Параметри	Тип	Обов'язковий	Опис
chat_id	Integer або String	Так	Унікальний ідентифікатор цільового чату або імені цільового каналу.
Photo	InputFile або String	Так	Фото для відправки. Можна передати file_id як рядок, щоб відправити фотографію яка вже знаходиться на серверах Telegram, або завантажити нову фотографію.
Параметри	Тип	Обов'язковий	Опис
Caption	String	Ні	Тема фотографії, від 0 до 200 символів.
reply_to_message_id	Integer	Ні	Ідентифікатор вихідного повідомлення.

Метод editMessageText використовується для редагування текстових повідомлень, які були відправлені ботом або через бота. Параметри методу наведені в таблиці 5.6.6.

Продовження таблиці 5.6.5.

Параметри	Тип	Обов'язковий	Опис
Replay_markup	InlineKeyboardMarkup or ReplyKeyboardMarkup or Hide or ForceReply	Ні	Розширені можливості пошуку інтерфейсу. Являє собою JSON-серіалізований об'єкт для вбудованої клавіатури, яка призначена для користувача, певні інструкції які використовуються, для приховування клавіатури користувача або примусової відповіді від користувача.

Таблиця 5.6.6 – Параметри методу editMessageText

Параметри	Тип	Обов'язковий	Опис
chat_id	Integer або String	Ні	Необхідний параметр, якщо inline_message_id не вказано. Унікальний ідентифікатор цільового чату або імені каналу.
Параметри	Тип	Обов'язковий	Опис
inline_message_id	String	Ні	Обов'язково, якщо chat_id і message_id не вказані. Ідентифікатор вбудованого повідомлення

Продовження таблиці 5.6.6.

Параметри	Тип	Обов'язковий	Опис
Text	String	Так	Новий текст повідомлення.
parse_mode	String	Ні	Потрібно надіслати Markdown або HTML, щоб додатки Telegram відображали напівжирний, курсивний, текст з фіксованою шириною або вбудовані URL-адреси в повідомленні нашого бота.
disable_web_page_preview	Boolean	Ні	Відключає попередній перегляд посилань для посилань в цьому повідомленні.
replay_markup	InlineKeyboardMarkup або ReplyKeyboardMarkup або ReplyKeyboardHide	Ні	Розширені можливості пошуку інтерфейсу.

Об'єкт типу User надає інформацію про користувача Telegram. Поля типу наведені в таблиці 5.6.7

Таблиця 5.6.7 – Поля об'єкта User

Поле	Тип	Опис
Id	Integer	Унікальний ідентифікатор бота або користувача
first_name	String	Ім'я бота або користувача
last_name	String	Не обов'язково. Прізвище бота або користувача
Username	String	Не обов'язково. Username бота або користувача

Об'єкт типу Chat являє собою інформацію про чат. Поля типу наведені в таблиці 5.6.8

Таблиця 5.6.8 – Поля об'єкта Chat

Поле	Тип	Опис
Id	Integer	Унікальний ідентифікатор чату. Абсолютне значення не перевищує $1e13$
Type	Enum	Тип чату: а) private; б) group; в) supergroup; г) channel;
Title	String	Не обов'язково. Назва, для каналів або груп
Username	String	Не обов'язково. Username, для чатів і деяких каналів
first_name	String	Не обов'язково. Ім'я співрозмовника в чаті
last_name	String	Не обов'язково. Прізвище співрозмовника в чаті
all_members_are_administrators	Boolean	Не обов'язково. True, якщо всі учасники чату є адміністраторами.

Об'єкт типу Message являю собою інформацію про повідомлення. Поля типу наведені в таблиці 5.6.9.

Таблиця 5.6.9 – Поля об'єкта Message

Поле	Тип	Опис
message_id	Integer	Унікальний ідентифікатор повідомлення
From	User	Не обов'язково. Відправник. Може бути порожнім в каналах.
Date	Integer	Дата відправлення повідомлення (Unix time)
Chat	Chat	Діалог, в якому було відправлено повідомлення
forward_from	User	Не обов'язково. Для пересланих повідомлень: відправник оригінального повідомлення
forward_date	Integer	Не обов'язково. Для пересланих повідомлень: дата відправки повідомлення.
reply_to_message	Message	Не обов'язково. Для відповідей: оригінальне повідомлення. Варто звернути увагу, що об'єкт Message в цьому полі не буде містити додаткові поля reply_to_message, навіть якщо він сам є відповіддю.
Text	String	Не обов'язково. Для текстових повідомлень: текст повідомлення, 0-4096 символів.

Продовження таблиці 5.6.9.

Поле	Тип	Опис
Entities	Масив з MessageEntity	Не обов'язково. Для текстових повідомлень: особливі суті в тексті повідомлення.
Document	Document	Не обов'язково. Інформація про фото
Photo	масив з PhotoSize	Не обов'язково. Доступні розміри фото
Sticker	Sticker	Не обов'язково. Інформація про стікери
Video	Video	Не обов'язково. Інформація про відеозапис
Voice	Voice	Не обов'язково. Інформація про голосове повідомлення
Caption	String	Не обов'язково. Підпис до файлу, фото або відео, 0-200 символів
Contact	Contact	Не обов'язково. Інформація про відправлений контакт.
Location	Location	Не обов'язково. Інформація про місцезнаходження.
Venue	Venue	Не обов'язково. Інформація про місце на карті.
new_chat_member	User	Не обов'язково. Інформація про користувача, доданому до групи.
left_chat_member	User	Не обов'язково. Інформація про користувача, якого видалили.

Продовження таблиці 5.6.9.

Поле	Тип	Опис
new_chat_title	String	Не обов'язково. Назва групи було змінено на це поле.
new_chat_photo	масив з PhotoSize	Не обов'язково. Фото групи було змінено на це поле.
Поле	Тип	Опис
group_chat_created	True	Не обов'язково. Сервісне повідомлення: група створена
supergroup_chat_created	True	Не обов'язково. Сервісне повідомлення: супергрупа створена.
channel_chat_created	True	Не обов'язково. Сервісне повідомлення: канал створений.
migrate_to_chat_id	Integer	Не обов'язково. Група була перетворена в супергрупу із зазначеним ідентифікатором. Не перевищує 1e13
migrate_from_chat_id	Integer	Не обов'язково. Супергрупа була створена з групи із зазначеним ідентифікатором. Не перевищує 1e13.
pinned_message	Message	Не обов'язково. Зазначене повідомлення було прикріплене. Варто звернути увагу, що об'єкт Message в цьому полі не буде містити додаткові поля reply_to_message, навіть якщо він сам є відповіддю

Об'єкт типу ReplyKeyboardMarkup являє собою клавіатуру з опціями відповіді. Поля типу наведені в таблиці 5.6.10.

Таблиця 5.6.10 – Поля об'єкта ReplyKeyboardMarkup

Поле	Тип	Опис
Keyboard	масив масивів з KeyboardButton	Масив рядів кнопок, кожен з яких є масивом об'єктів KeyboardButton
resize_keyboard	Boolean	Не обов'язково. Вказує клієнту підігнати висоту клавіатури під кількість кнопок (зробити її менше, якщо кнопок мало). За замовчуванням False, тобто клавіатура завжди такого ж розміру, як і стандартна клавіатура пристрою.
one_time_keyboard	Boolean	Не обов'язково. Вказує клієнту приховати клавіатуру після використання (після натискання на кнопку). Її як і раніше можна буде відкрити через іконку в поле вводу повідомлення. За замовчуванням False.
Selective	Boolean	Не обов'язково. Цей параметр потрібен, щоб показувати клавіатуру тільки деяким користувачам. Приклад: користувач відправляє запит на зміну мови бота. Бот відправляє клавіатуру зі списком мов, видиму тільки цьому користувачеві.

Об'єкт типу `KeyboardButton` є ще однією кнопкою в клавіатурі відповіді. Для звичайних текстових кнопок цей об'єкт може бути змінений на рядок, що містить текст на кнопці. Поля типу наведені в таблиці 5.6.11.

Таблиця 5.6.11 – Поля об'єкта `KeyboardButton`

Поле	Тип	Опис
Text	String	Текст на кнопці. Якщо жодне з опціональних полів не використано, то при натисканні на кнопку цей текст буде відправлений боту як просте повідомлення.
request_contact	Boolean	Не обов'язково. Якщо значення <code>True</code> , то при натисканні на кнопку боту відправиться контакт користувача з його номером телефону. Доступно тільки в діалогах з ботом.
request_location	Boolean	Не обов'язково. Якщо значення <code>True</code> , то при натисканні на кнопку боту відправиться розташування користувача. Доступно тільки в діалогах з ботом.

Об'єкт типу `InlineKeyboardMarkup` є вбудовану клавіатуру, яка з'являється під відповідним повідомленням. Поля типу наведені в таблиці 5.6.12.

Об'єкт типу `InlineKeyboardButton` це ще одна кнопка вбудованої клавіатури. Ми обов'язково повинні задіяти рівно одне опціональне поле. Поля типу наведені в таблиці 5.6.13.

Таблиця 5.6.12 – Поля об'єкта InlineKeyboardMarkup

Поле	Тип	Опис
inline_keyboard	Масив масивів з InlineKeyboardButton	Масив рядків, кожна з яких є масивом об'єктів InlineKeyboardButton.

Таблиця 5.6.13 – Поля об'єкта Inline KeyboardButton

Поле	Тип	Опис
Text	String	Текст на кнопці
url	String	Не обов'язково. URL, який відкриється при натисканні на кнопку
callback_data	String	Не обов'язково. Дані, які будуть відправлені в callback_query при натисканні на кнопку.
switch_inline_query	String	Не обов'язково. Якщо цей параметр заданий, то при натисканні на кнопку додаток запропонує користувачеві вибрати будь-який чат, відкриє його і вставить в поле вводу повідомлення юзернейм бота і певний запит для вбудованого режиму. Якщо відправляти пусте поле, то буде вставлений тільки юзернейм бота.

Продовження таблиці 5.6.13

Поле	Тип	Опис
switch_inline_query_current_chat	String	Не обов'язково. Якщо встановлено, натискання кнопки введе ім'я бота і вказаний вбудований запит в поле вводу поточного чату. Може бути порожнім, і в цьому випадку буде вставлено тільки ім'я користувача бота.
callback_game	CallbackGame	Не обов'язково Опис гри, яка буде запущена, коли користувач натисне кнопку. Примітка: Цей тип кнопки завжди повинен бути першою кнопкою в першому рядку.

6 БАЗА ДАНИХ – MONGODB

MongoDB – це база даних, яка є документо-орієнтована з відкритим вихідним кодом і вона не вимагає опису схем таблиць. Головні риси та можливості:

- а) підтримка відмовостійкості і масштабованості;
- б) асинхронна реплікація, набір реплік і розподілу бази даних на вузли;
- в) JSON-подібна схема зберігання даних;
- г) ефективне зберігання великих об'єктів, адміністративний інтерфейс, серверні функції, Map / Reduce і інше;
- д) використання Javascript в якості мови для формування запитів;
- е) профілювання запитів;
- ж) широкий набір атомарних операцій над даними (умовний пошук, складна вставка / оновлення тощо);
- з) різні типи даних (в тому числі підтримка масивів); підтримка індексів (B-Tree);
- и) повнотекстовий пошук, в тому числі українською мовою, з підтримкою морфології;
- к) журнал операцій, що змінюють дані в базі даних.

MongoDB пропонує документо-орієнтовану модель даних на відміну від реляційних баз даних, завдяки цьому вона працює в рази швидше, має кращу масштабованість та її легше використовувати. Однак, враховуючи всі недоліки традиційних баз даних і переваги MongoDB, важливим розумінням є те, що завдання бувають різні і методи їх вирішення бувають також різні.

В певній ситуації MongoDB дійсно здатна поліпшити продуктивність програми, для прикладу візьмемо ситуацію, що існує потреба зберігати складні за структурою дані. В іншій же ситуації краще буде використовувати традиційні реляційні бази даних.

MongoDB може представляти не тільки єдину базу даних, що знаходиться на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати декілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність.

В процесі створення телеграм-бота була спроектована схема взаємодії користувача з базою даних, яка дає відповіді на його повідомлення (рисунок 6.1).

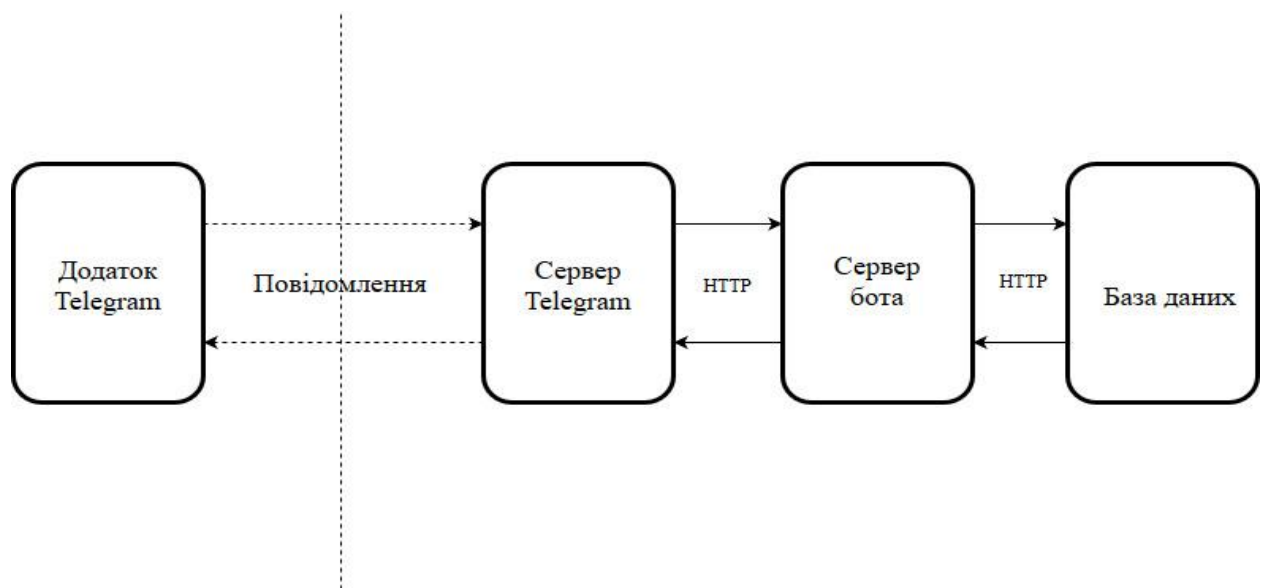


Рисунок 6.1 – Схема взаємодії користувача з базою даних

Ось кілька важливих властивостей цієї бази даних:

а) формат даних в MongoDB. Одним з популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні за структурою дані. Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON або скорочення від binary JSON. BSON дозволяє працювати з даними швидше: швидше виконується пошук і обробка. Хоча треба зазначити, що BSON на відміну від зберігання даних в форматі JSON має невеликий недолік: в цілому дані в JSON- форматі займають менше місця, ніж в форматі BSON, з іншого боку, даний недолік з лишком окупається швидкістю;

б) кросплатформеність. MongoDB написана на C ++, тому її легко перенести на різні платформи. MongoDB може бути розгорнута на платформах Windows, Linux, MacOS, Solaris. Можна також завантажити вихідний код і самому скомпілювати MongoDB, але рекомендується використовувати бібліотеки з офіційного сайту;

в) колекції. Якщо в традиційному світі SQL є таблиці, то в світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити найрізноманітніші об'єкти, що мають різну структуру і різний набір властивостей;

г) реплікація. Система зберігання даних в MongoDB представляє набір реплік. У цьому наборі є основний вузол, а також може бути набір вторинних вузлів. Всі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один з вторинних вузлів стає головним;

д) простота у використанні. Відсутність жорсткої схеми бази даних і в зв'язку з цим потреби при щонайменшій зміні концепції зберігання даних створювати знову цю схему значно полегшують роботу з базами даних MongoDB і подальшим їх масштабуванням. Крім того, економиться час розробників. Їм більше не треба думати про нове створення бази даних і витрачати час на побудову складних запитів;

е) GridFS. Однією з проблем при роботі з будь-якими системами баз даних є збереження даних великого розміру. Можна зберігати дані в файлах, використовуючи різні мови програмування. Деякі СУБД пропонують спеціальні типи даних для зберігання бінарних даних в БД (наприклад, BLOB в MySQL). На відміну від реляційних СУБД MongoDB дозволяє зберігати різні документи з різним набором даних, однак при цьому розмір документа обмежується 16 Мб. Але MongoDB пропонує рішення - спеціальну технологію GridFS, яка дозволяє зберігати дані за розміром більше, ніж 16 Мб. Система GridFS складається з двох колекцій. У першій колекції, яка

називається files, зберігаються імена файлів, а також їх метадані, наприклад, розмір. А в іншій колекції, яка називається chunks, у вигляді невеликих сегментів зберігаються дані файлів, зазвичай сегментами по 256 Кб.

					ІТ51.320БАК.002 ПЗ	Лист
						49
Ізм.	Лист	№ докум.	Підпис	Дата		

7 TELEGRAM

Telegram – сучасний популярний сервіс для обміну інформацією. Крім версії для браузерів існують також версії для ОС Linux, Mac та Windows, а також для популярних мобільних ОС, таких як Android, iOS та Windows Phone.

Однією з головних функціональних можливостей цього сервісу є можливість створення сторонніх додатків, так званих «ботів», які можуть взаємодіяти з користувачами, відповідати на будь-які команди користувача, а також відправляти їм повідомлення. Наприклад, вже сьогодні існує безліч ботів для отримання новин, останнього прогнозу погоди, для створення заміток і нагадувань, для проведення банківських операцій, тощо.

Повідомлення, команди і запити, надіслані користувачами, передаються на сервер, а саме до розробників пошукових роботів. Сервер Telegram є тільки посередником і надає свій API для розробників, доступний за допомогою HTTPS-інтерфейсу, який пропонує спрощену версію API Telegram, так званий «Bot API». Для того, щоб почати взаємодіяти з ботом, користувач повинен спочатку додати його до себе в чат або в групу. або відправляючи команди з чату безпосередньо якомусь боту, використовуючи символ «@» перед його ім'ям в повідомленні з командою. Детальніше про боти можна почитати на офіційному сайті Telegram, в даній роботі нас цікавить тільки можливість додавання бота в чат або групу, після чого він зможе надсилати туди оповіщення при будь-яких відхиленнях аналізованих метрик.

Слід уточнити етап створення нового бота і використання API, що надається Telegram. Згідно інструкції з офіційного сайту, для створення бота мінімально необхідні такі пункти:

а) додати бота «BotFather» до себе в чат, який дозволяє керувати створенням і налаштуванням нових ботів всередині Telegram;

					IT51.320БАК.002 ПЗ	Лист
						50
Ізм.	Лист	№ докум.	Підпис	Дата		

б) надіслати боту команду «/newbot», після чого той запросить повне і коротке ім'я створюваного бота (повне ім'я використовується для відображення користувачеві назви бота в заголовку чату, коротке ім'я для посилення швидких команд) і згенерує унікальний токен авторизації, який буде потрібен для ідентифікації бота і взаємодії через TelegramAPI при відправці запитів.

					IT51.320БАК.002 ПЗ	Лист
						51
Ізм.	Лист	№ докум.	Підпис	Дата		

8 ТЕСТУВАННЯ СИСТЕМИ

Тестування телеграм бота здійснювалося вручну. Після кожного етапу розробки виконувалося тестування працездатності по заздалегідь підготовленим тест-кейсам, які були складені при визначені цілей розробки і інструментів, які використовувалися.

Робота з Telegram ботом:

- а) швидкість відгуку бота на повідомлення про початок роботи;
- б) коректна обробка повідомлень різного типу від користувача.
- в) коректне відображення діалогу;
- г) обробка натискання на клавішу відправки повідомлення;
- д) швидкість відображення відповіді після того, як користувач, написав повідомлення;
- е) швидкість роботи бази даних;
- ж) оцінювання коректності відповідей.

Тестування боту проводилося на мобільному пристрої Apple Iphone 7 Plus з наступними технічними характеристиками.

- а) екран: 5,5 ", IPS LCD, 1920x1080, мультитач;
- б) процесор: Apple A10 Fusion, 4x1;
- в) операційна система: iOS 12.1.4;
- г) оперативна пам'ять: 3 ГБ;
- д) вбудована пам'ять: 128 ГБ;
- е) навігація: GPS;
- ж) Telegram: Telegram v 5.7.

Після приєднання до чату користувачу доступна тільки одна команда – start. Відправивши цю команду, в відповідь отримуємо основну інформацію про бота (рисунок 8.1).

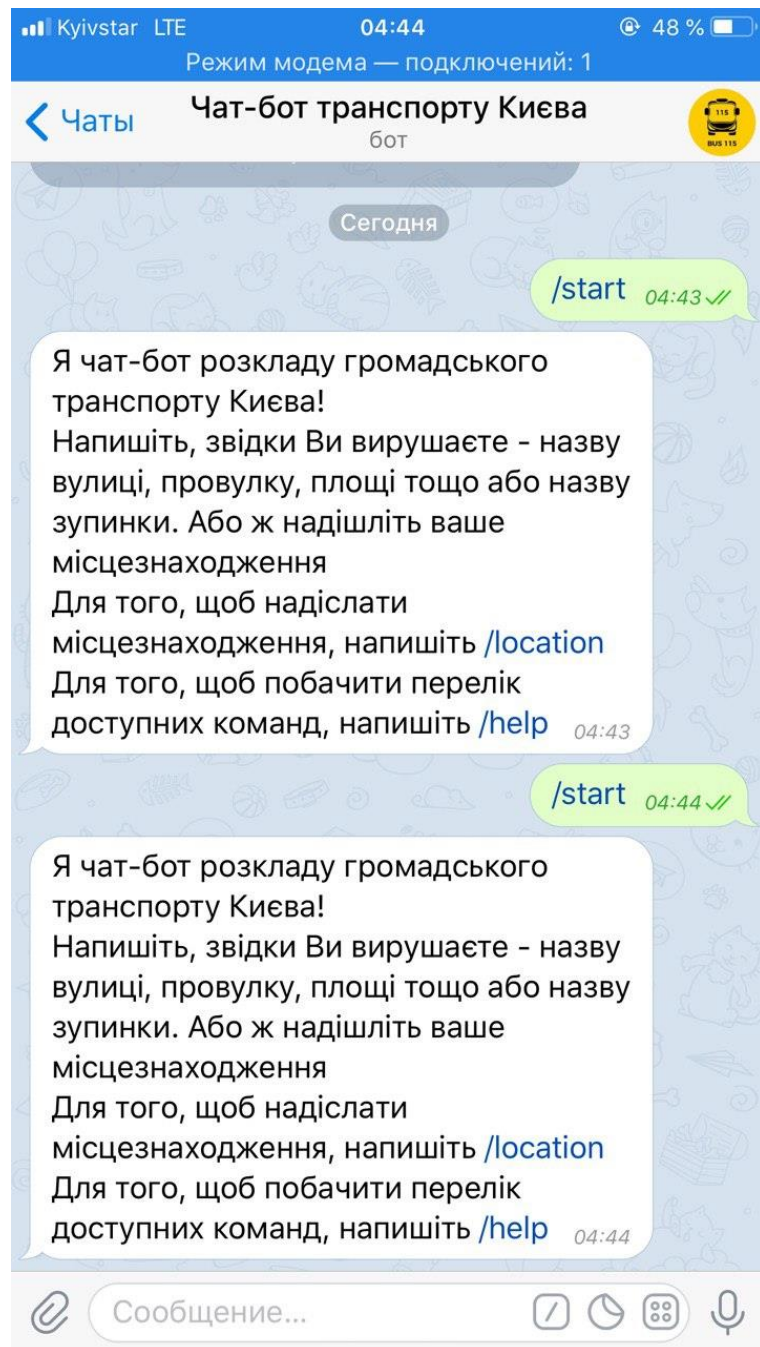


Рисунок 8.1 – Початок роботи з ботом

Після цього у нас з'являється можливість надіслати своє місцезнаходження, натиснувши кнопку Location або відправивши команду /location. В свою чергу бот в цілях безпеки запитає дозвіл на отримання інформації про місцезнаходження користувача (рисунок 8.2).

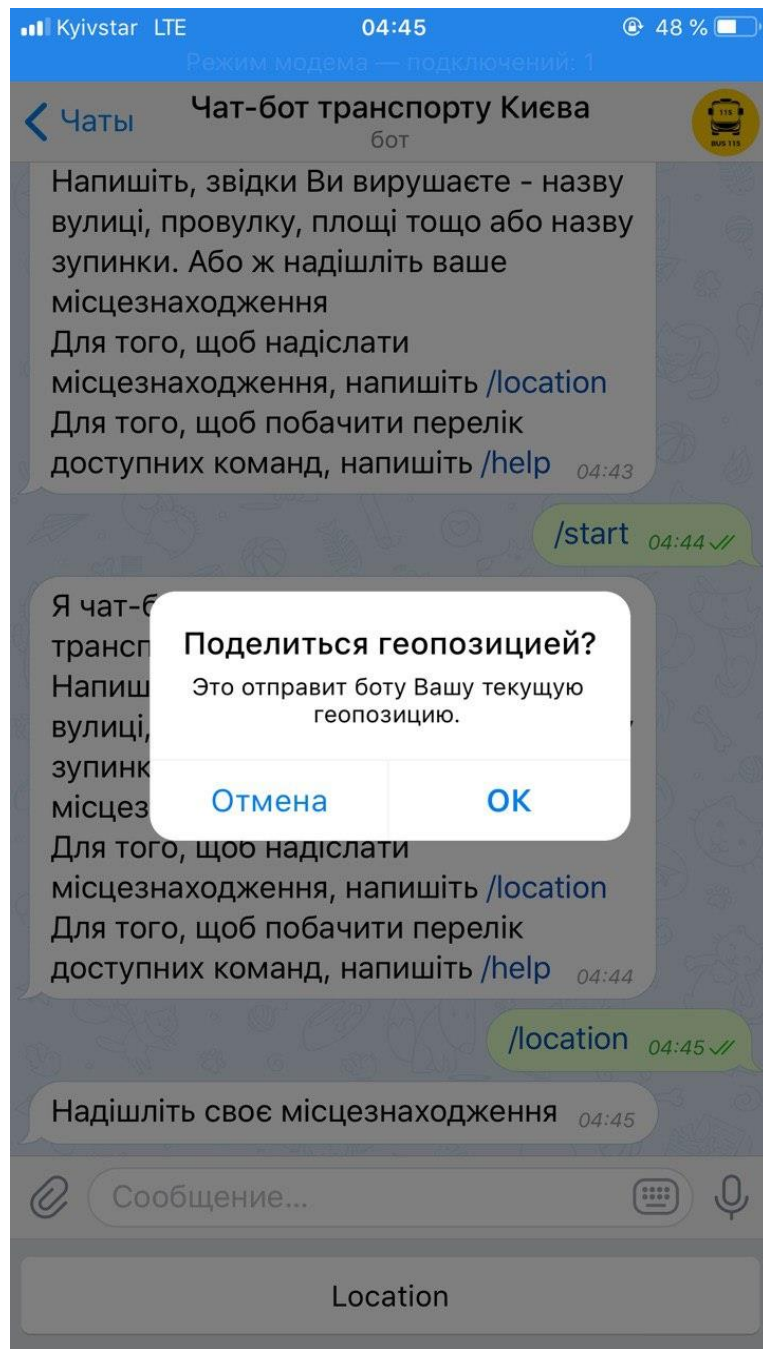


Рисунок 8.2 – Запит на отримання доступу до геопозиції

Як тільки користувач дасть згоду на отримання ботом геопозиції, бот автоматично буде шукати всі можливі зупинки громадського транспорту в радіусі 500 метрів.

Ввівши адресу, яка потрібна користувачеві, бот буде шукати всі можливі зупинки громадського транспорту і відправити всі знайдені варіанти в чат (рисунок 8.3).

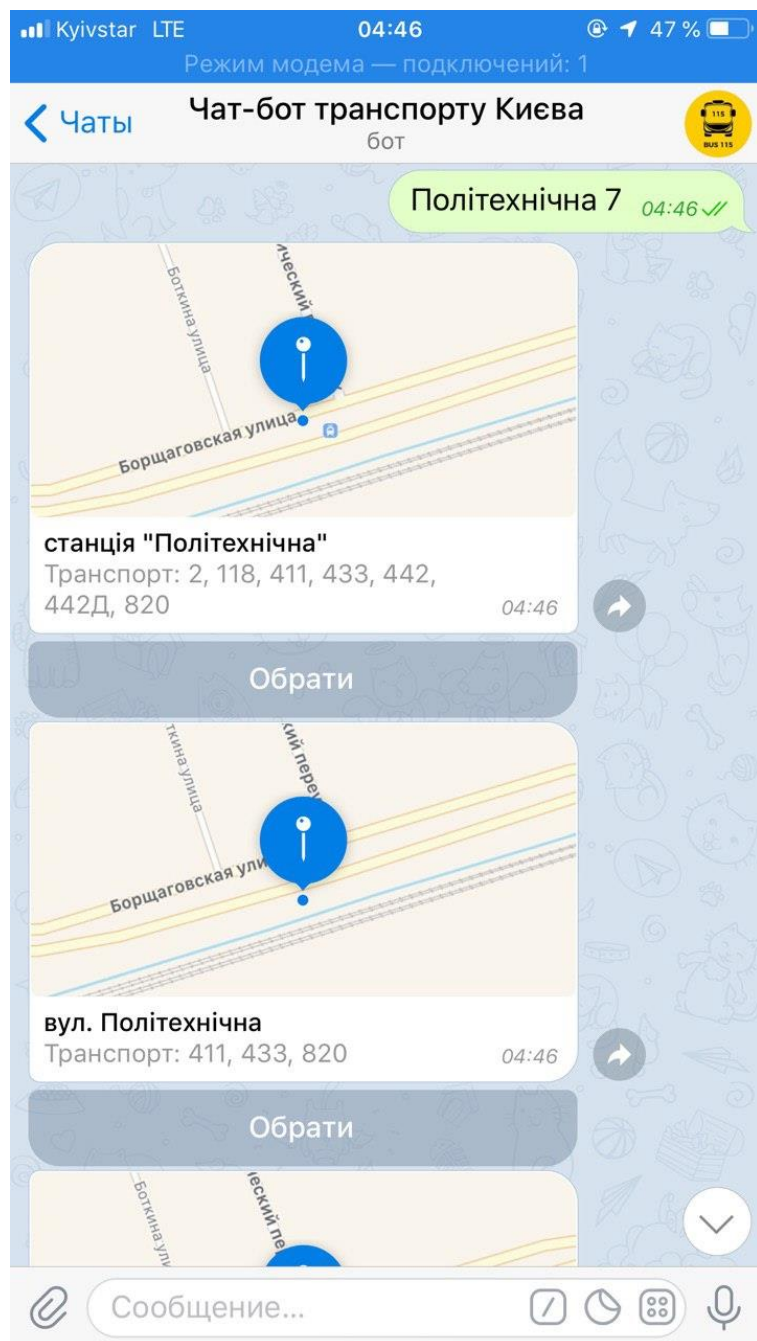


Рисунок 8.3 – Відповідь бота на запит

Також бот надає можливість завідома бачити, який громадський транспорт зупиняється на тій, чи іншій зупинці.

При виборі зупинки громадського транспорту ми отримуємо інформацію про очікуваний час прибуття громадського транспорту (рисунок 8.4).

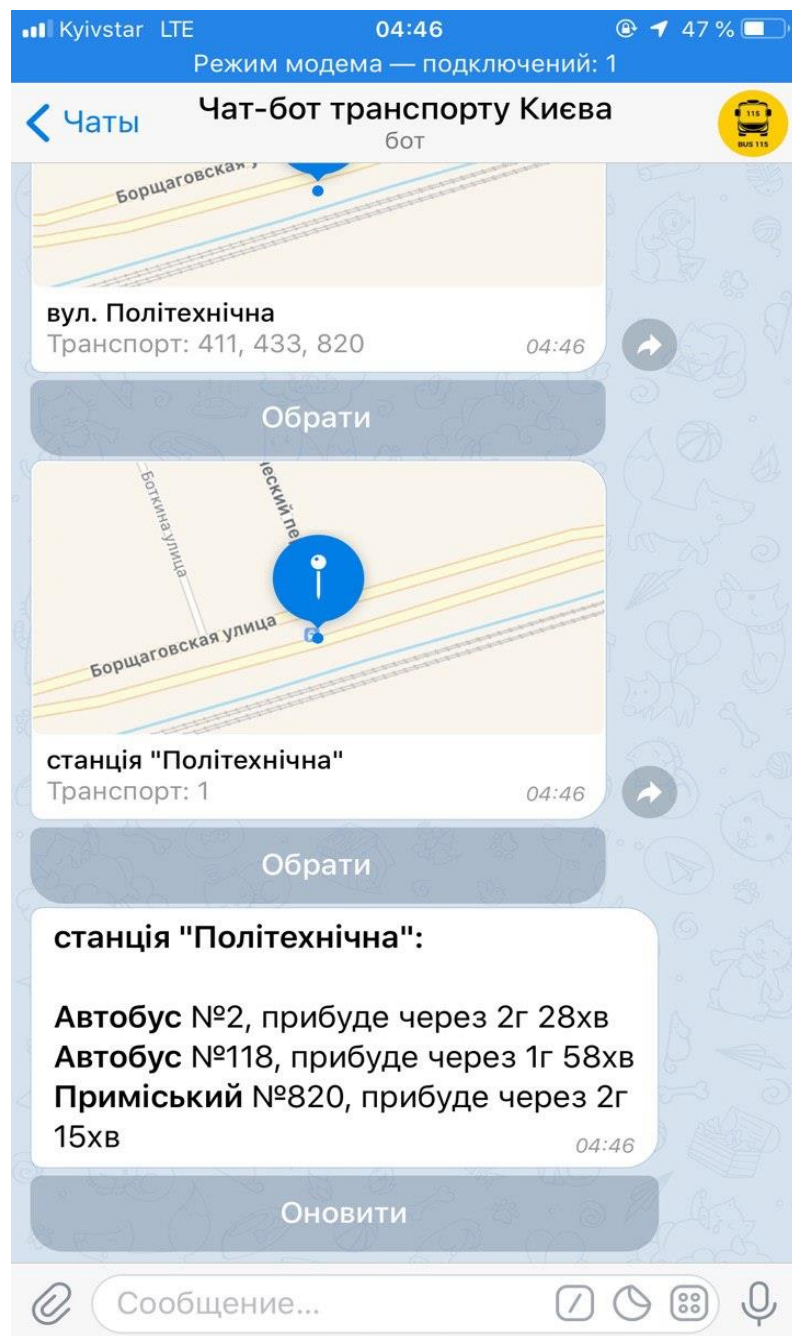


Рисунок 8.4 – Детальна інформація про зупинку

9 ОПИС ВЛАСНОЇ РЕАЛІЗАЦІЇ

Відповідно до завдання дипломного проекту та відповідно до розглянутих підходів було вирішено за допомогою засобів TeleBot, CherryPy та Requests на мові програмування Python створити телеграм-бота, який буде надавати необхідну інформацію.

Основні аспекти які планується реалізувати в ході виконання дипломного проекту:

- а) отримання запита користувача;
- б) обробка запита користувача;
- в) за необхідністю уточнити деякі аспекти запиту;
- г) відповідно до обробленого запиту видати необхідну інформацію користувачеві;
- д) передбачити реакцію боту на непередбачені запити;

Також за можливості розглянути та реалізувати такі аспекти:

- а) створення бази даних громадського транспорту та зупинок;
- б) отримання координат місцезнаходження користувача;
- в) запропонувати декілька варіантів зупинок та маршрутів.

Більш детально опишемо кожен з аспектів. Ми впевнені, що користувач захоче задати своє запитання на природній мові, або вибрати з запропонованих нами варіантів, далі це повідомлення буде передаватися у сервіс за допомогою якого ми зможемо отримати необхідний нам контекст повідомлення, потім ця інформацію буде передаватися до нашої програми, в якій ми будемо визначати що робити з цим повідомленням, відразу надавати інформацію, або потрібно задати певні уточнюючі питання, процес уточнення може повторюватися доки ми не отримаємо необхідну інформацію для видачі максимально повної відповіді для користувача, після завершення процесу отримання запиту користувача та його аналізування, ми надсилаємо йому необхідну інформацію.

Якщо користувач ввів непередбачений нами запит або запит який не стосується нашої предметної області, то є два варіанти розвитку. Якщо запит не стосується теми, то буде виводитися повідомлення про те щоб користувач вводив питання тільки стосовно транспорту. Якщо ж користувач ввів запит який стосується транспорту, але бот за якоїсь причини або не зрозумів його, або ми не змогли надати необхідну інформацію щодо цього запиту, ми будемо відповідати заздалегідь підготовленими відповідями.

					ІТ51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		58

ВИСНОВКИ

В ході виконання дипломної роботи було проведено дослідження предметної області, визначено головні вимоги до системи та бізнес-процеси. Проаналізовано вимоги до системи в цілому, вимоги до функцій системи, програмного і технічного забезпечення.

Було проведено дослідження технологій для побудови телеграм-бота. В результаті дослідження були обрані наступні мови програмування та технології: Python, модуль TeleBot, модуль CherryPy, модуль Requests, Telegram API та база даних MongoDB .

Результатом проведеного дослідження стало створення телеграм-бота для побудови маршрутів з використанням громадського транспорту міста Києва. Система є сучасною та зручною, задовольняє всі вимоги з точки зору функціональності, юзабіліті, логіки.

					IT51.320БАК.002 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		59

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Подборка: 30 полезных ботов [Електронний ресурс] – Режим доступу: <https://ain.ua/2017/08/14/30-telegram-botov/>
2. Telegram Bot API [Електронний ресурс] – Режим доступу: <https://core.telegram.org/bots/api>
3. Самоучитель Python [Електронний ресурс] : Режим доступу: <https://pythonworld.ru/samouchitel-python>
4. Telegram FAQ [Електронний ресурс]. – Режим доступу: <https://telegram.org/faq>
5. Сайт Eway [Електронний ресурс]. – Режим доступу: <https://www.eway.in.ua/ru/cities/kyiv>
6. Telegram-bot[Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/442800/>
7. Телеграм бот за допомогою TeleBot [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/448310/>
8. Бот для Telegram на Python. Heroku сервер [Електронний ресурс]. – Режим доступу: <https://tproger.ru/translations/telegram-bot-create-and-deploy/>
9. CherryPy [Електронний ресурс]. – Режим доступу: <https://cherrypy.org/>
10. Створювання бота. Telebot [Електронний ресурс]. – Режим доступу: <https://medium.com/@plug2up/%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B5%D0%BC-%D1%82%D0%B5%D0%BB%D0%B5%D0%B3%D1%80%D0%B0%D0%BC-%D0%B1%D0%BE%D1%82%D0%B0-2c8377d4d482>
11. Python Requests [Електронний ресурс]. – Режим доступу: <https://2.python-requests.org/en/master/>
12. Руководство по работе с HTTP в Python. Библиотека requests [Електронний ресурс]. – Режим доступу: <https://khashtamov.com/ru/python-requests/>

13. Сайт Heroku [Електронний ресурс]. – Режим доступу:
<https://www.heroku.com/>
14. Heroku – Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://ru.wikipedia.org/wiki/Heroku>
15. Огляд протоколу HTTP [Електронний ресурс]. – Режим доступу:
<https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
16. Черга повідомлень [Електронний ресурс]. – Режим доступу:
<https://aws.amazon.com/ru/message-queue/>
17. Synbot [Електронний ресурс]. – Режим доступу: https://ru.telegram-store.com/catalog/bots/synbot_/
18. REST API[Електронний ресурс]. – Режим доступу:
<https://habr.com/ru/post/351890/>
19. JSON[Електронний ресурс]. – Режим доступу:
<https://www.json.org/json-ru.html>
20. Тестування телеграм-бота[Електронний ресурс]. – Режим доступу:
<https://habr.com/ru/post/322816/>
21. What is Artificial Intelligence (AI) [Електронний ресурс]. – Режим доступу: <https://www.techopedia.com/definition/190/artificial-intelligence-ai>